

FILEID**PRINT

H 1

PPPPPPPP	RRRRRRRR		NN	NN	TTTTTTTT
PPPPPPPP	RRRRRRRR		NN	NN	TTTTTTTT
PP	PP	RR RR		NN	NN TT
PP	PP	RR RR		NN	NN TT
PP	PP	RR RR		NNNN	NN TT
PP	PP	RR RR		NNNN	NN TT
PPPPPPPP	RRRRRRRR		NN NN	NN	TT
PPPPPPPP	RRRRRRRR		NN NN	NN	TT
PP	RR RR		NN	NNNN	TT
PP	RR RR		NN	NNNN	TT
PP	RR RR		NN	NN	TT
PP	RR RR		NN	NN	TT
PP	RR RR		NN	NN	TT
PP	RR RR		NN	NN	TT
LL			SSSSSSSS		
LL			SSSSSSSS		
LL			SS		
LL			SS		
LL			SS		
LL			SSSSSS		
LL			SSSSSS		
LL			SS		
LL			SS		
LL			SS		
LL			SS		
LLLLLLLL			SSSSSSSS		
LLLLLLLL			SSSSSSSS		

1 0001 0 MODULE MAIL\$PRINT(%TITLE 'QUEUE MAIL MESSAGE FOR PRINTING'
2 0002 0 IDENT = 'V04-000') =
3 0003 1 BEGIN
4 0004 1
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 **
31 0031 1 FACILITY: VAX/VMS MAIL UTILITY
32 0032 1 ABSTRACT: QUEUE THE CURRENT MESSAGE FOR PRINTING
33 0033 1 ENVIRONMENT: NATIVE/USER MODE
34 0034 1 AUTHOR: LEN KAWELL, CREATION DATE: 10-JUN-1979
35 0035 1
36 0036 1
37 0037 1
38 0038 1
39 0039 1
40 0040 1
41 0041 1
42 0042 1
43 0043 1
44 0044 1
45 0045 1
46 0046 1
47 0047 1
48 0048 1
49 0049 1
50 0050 1
51 0051 1
52 0052 1
53 0053 1
54 0054 1
55 0055 1
56 0056 1
57 0057 1
MODIFIED BY:
V03-008 ROP0019 Robert Posniak 6-JUL-1984
Again - Unless /MAIL, use process default protection on
EXTRACT.
V03-007 BLS0272 Benn Schreiber 18-FEB-1984 13:38:20
Sigh. Missing dot.
V03-006 BLS0263 Benn Schreiber 5-FEB-1984
Unless /MAIL, use process default protection on EXTRACT.
V03-005 BLS0255 Benn Schreiber 28-Dec-1983
Include extract command code in here. Add extract/mail
to create sequential mail file. When printing, set
flag_first and no-trail if we can.
V03-004 BLS0227 Benn Schreiber 6-Jul-1983
Upgrade to latest job controller

MAIL\$PRINT
VO4-000

QUEUE MAIL MESSAGE FOR PRINTING

J 1
16-Sep-1984 01:14:12 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:42:29 DISK\$VMSMASTER:[MAIL.SRC]PRINT.B32;1 Page 2
(1)

58	0058	1	
59	0059	1	V03-003 BLS0225 Benn Schreiber 17-Jun-1983
60	0060	1	Set FOP=SQ0 on temp file
61	0061	1	
62	0062	1	V03-002 BLS0222 Benn Schreiber 19-May-1983
63	0063	1	Make temp file in SYSS\$SCRATCH:
64	0064	1	
65	0065	1	V03-001 BLS0211 Benn Schreiber 7-Jan-1983
66	0066	1	Add support for ISAM mail files. Recursively translate
67	0067	1	queue name.
68	0068	1	--

```

70 0069 1
71 0070 1 | INCLUDE FILES:
72 0071 1
73 0072 1 | LIBRARY 'SYSSLIBRARY:STARLET';
74 0073 1 | REQUIRE 'SRC$MAILREQ';
75 0219 1 | LIBRARY 'LIB$MAILDEF';
76 0220 1
77 0221 1 | TABLE OF CONTENTS:
78 0222 1
79 0223 1 | FORWARD ROUTINE
80 0224 1 | MAIL$PRINT: NOVALUE,
81 0225 1 | CANCEL_GETQUI : NOVALUE,
82 0226 1 | CLOSE_JOB : NOVALUE;
83 0227 1
84 0228 1 | OWN
85 0229 1 | CURQUEUE DESC : DYN_DESC DECL,
86 0230 1 | DEFDIRDESC : DYN_DESC DECL,
87 0231 1 | CURJOBID,
88 0232 1 | FILES QUEUED,
89 0233 1 | EXITSTATUS: LONG,
90 0234 1 | GETQUIEXITBLOCK: VECTOR[4]
91 0235 1 | INITIAL(0,
92 0236 1 | | CANCEL_GETQUI,
93 0237 1 | | 1,
94 0238 1 | | EXITSTATUS),
95 0239 1 | EXITBLOCK: VECTOR[4]
96 0240 1 | INITIAL(0,
97 0241 1 | | CLOSE_JOB,
98 0242 1 | | 1,
99 0243 1 | | EXITSTATUS),
100 0244 1 | EXITDECLARED: LONG;
101 0245 1
102 0246 1 | EXTERNAL REFERENCES:
103 0247 1
104 0248 1 | EXTERNAL
105 0249 1 | MAIL$GL FLAGS : $BBLOCK,
106 0250 1 | MAIL$L_NMSG,
107 0251 1 | MAIL$L_CURMSG: REF $BBLOCK,
108 0252 1 | MAIL$L_MSG_NJUM,
109 0253 1 | MAIL$L_MSG_STS,
110 0254 1 | MAIL$L_MSG_RADR,
111 0255 1 | MAIL$W_MSG_RSIZ : WORD,
112 0256 1 | MAIL$L_MSGBUFSIZ,
113 0257 1 | MAIL$Q_USERDEF : $BBLOCK,
114 0258 1 | MAIL$B_MSGTRAN : BYTE;
115 0259 1
116 0260 1 | EXTERNAL ROUTINE
117 0261 1 | CLISPRESENT,
118 0262 1 | CLISGET VALUE,
119 0263 1 | MAIL$FIND_MSG,
120 0264 1 | MAIL$FIND_I_MSG,
121 0265 1 | MAIL$GET_I_MSG,
122 0266 1 | MAIL$GET_MSG,
123 0267 1 | MAIL$GETMSG,
124 0268 1 | MAIL$PUT_OUTPUT,
125 0269 1 | MAIL$REWIND_I_MSG,
126 0270 1 | MAIL$REWIND_MSG,

```

! mail control flags
 ! Number of messages currently in vector
 ! Pointer to current message block
 ! Current message number
 ! Message status
 ! Sequential msg address
 ! Sequential msg size
 ! Size of whole message buffer (seq file)
 ! Descriptor of mail dir
 ! Transition for isam msg reading
 ! True if qualifier present
 ! Get qualifier value
 ! Find sequential message
 ! Find isam msg
 ! Get ISAM message text
 ! Get message text
 ! Get text for signal code
 ! Put to SYSSOUTPUT
 ! Rewind isam message
 ! Rewind sequential message

```
127 0271 1 MAIL$TEST_SUBJ,  
128 0272 1 STR$COPY DX,  
129 0273 1 STR$CONCAT,  
130 0274 1 STR$FREE1 DX,  
131 0275 1 UTIL$GETFILENAME,  
132 0276 1 UTIL$REPORT_IO_ERROR;  
133  
134 0277 1 Define protection mask  
135 0279 1  
136 0280 1 LITERAL  
137 0281 1 PRO_SYS = $BITPOSITION(XAB$V_SYS),  
138 0282 1 PRO_OWN = $BITPOSITION(XAB$V_OWN),  
139 0283 1 PRO_GRP = $BITPOSITION(XAB$V_GRP),  
140 0284 1 PRO_WLD = $BITPOSITION(XAB$V_WLD),  
141 0285 1 NO_ACCESS = XAB$M_NOREAD OR XAB$M_NOWRITE OR XAB$M_NOEXE OR XAB$M_NODEL,  
142 0286 1  
143 0287 1 SND_PRO = ! Send file protection  
144 0288 1 ( XAB$M_NOEXE OR XAB$M_NODEL ) ^ PRO_SYS OR  
145 0289 1 ( XAB$M_NOEXE OR XAB$M_NODEL ) ^ PRO_OWN OR  
146 0290 1 (XAB$M_NOREAD OR XAB$M_NOWRITE OR XAB$M_NOEXE OR XAB$M_NODEL ) ^ PRO_GRP OR  
147 0291 1 (XAB$M_NOREAD OR XAB$M_NOWRITE OR XAB$M_NOEXE OR XAB$M_NODEL ) ^ PRO_WLD,  
148 0292 1  
149 0293 1 FILE_PRO =  
150 0294 1 (XAB$M_NOEXE ^ PRO_SYS) OR  
151 0295 1 (XAB$M_NOEXE ^ PRO_OWN) OR  
152 0296 1 (NO_ACCESS ^ PRO_GRP) OR  
153 0297 1 (NO_ACCESS ^ PRO_WLD);  
154 0298 1  
155 0299 1 Define local names for shared messages  
156 0300 1  
157 P 0301 1 $SHR_MSGDEF(MAIL,126,LOCAL,  
158 P 0302 1 (CREATED,INFO),  
159 P 0303 1 (CONFQUAL,WARNING),  
160 P 0304 1 (INQUAVAL,WARNING),  
161 P 0305 1 (OPENOUT,ERROR),  
162 P 0306 1 (CLOSEOUT,ERROR),  
163 P 0307 1 (READERR,WARNING),  
164 P 0308 1 (WRITEERR,ERROR));  
165 P 0309 1  
166 P 0310 1 BIND  
167 P 0311 1 FF_STRING = UPLIT(BYTE(FF)),  
168 P 0312 1 LISTFILE_DESC = $DESCRIPTOR('SYS$SCRATCH:MAIL.LIS') : $BBLOCK;  
169 P 0313 1  
170 P 0314 1 SD(  
171 P 0315 1 'APPEND',  
172 P 0316 1 'ALL',  
173 P 0317 1 'EXTF',  
174 P 0318 1 'HEADER',  
175 P 0319 1 'MAIL');
```

MAIL\$PRINT
V04-000

QUEUE MAIL MESSAGE FOR PRINTING

M 1
16-Sep-1984 01:14:12
14-Sep-1984 12:42:29
VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]PRINT.B32;1 Page 5
(3)

```
177 0320 1 ROUTINE WRITE_REC (SND_RAB) =  
178 0321 1 ++  
179 0322 1 FUNCTIONAL DESCRIPTION:  
180 0323 1  
181 0324 1 Write record to output file  
182 0325 1  
183 0326 1 INPUTS:  
184 0327 1  
185 0328 1 SND_RAB is address of rab, RSZ and RBF filled in  
186 0329 1 --  
187 0330 2 BEGIN  
188 0331 2  
189 0332 3 RETURN $PUT(RAB=.SND_RAB,ERR=UTIL$REPORT_IO_ERROR)  
190 0333 1 END;
```

```
.TITLE MAIL$PRINT QUEUE MAIL MESSAGE FOR PRINTING  
.IDENT \V04-000\  
.PSECT $CODE$,NOWRT,2  
49 41 4D 3A 48 43 54 41 52 43 53 24 53 59 53 0C 00000 P.AAA: .BYTE 12  
53 49 4C 2E 4C 00001 P.AAC: .ASCII \SYSSSCRATCH:MAIL.LIS\  
00015 .BLKB 3  
00000014 00018 P.AAB: .LONG 20  
00000000 0001C P.AAC: .ADDRESS P.AAC  
44 4E 45 50 50 41 00020 P.AAE: .ASCII \APPEND\  
00026 .BLKB 2  
00000006 00028 P.AAD: .LONG 6  
00000000 0002C P.AAG: .ADDRESS P.AAE  
4C 4C 41 00030 P.AAG: .ASCII \ALL\  
00033 .BLKB 1  
00000003 00034 P.AAF: .LONG 3  
00000000 00038 P.AAF: .ADDRESS P.AAG  
46 54 58 45 0003C P.AAI: .ASCII \EXTF\  
00000004 00040 P.AAH: .LONG 4  
00000000 00044 P.AAI: .ADDRESS P.AAI  
52 45 44 41 45 48 00048 P.AAK: .ASCII \HEADER\  
0004E .BLKB 2  
00000006 00050 P.AAJ: .LONG 6  
00000000 00054 P.AAJ: .ADDRESS P.AAK  
4C 49 41 4D 00058 P.AAM: .ASCII \MAIL\  
00000004 0005C P.AAL: .LONG 4  
00000000 00060 P.AAM: .ADDRESS P.AAM  
.PSECT $OWNS,NOEXE,2  
00# 00000 CURQUE_DESC: .BYTE 0[2]  
02 0E 00002 .BYTE 14, 2  
00004 .BLKB 4  
00# 00008 DEFDIRDESC: .BYTE 0[2]  
02 0E 0000A .BYTE 14, 2  
0000C .BLKB 4  
00010 CURJOBID:
```

MAIL\$PRINT
VO4-000

QUEUE MAIL MESSAGE FOR PRINTING

N 1
16-Sep-1984 01:14:12 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:42:29 DISK\$VMSMASTER:[MAIL.SRC]PRINT.B32;1 Page 6
(3)

00014 FILES_QUEUED: .BLKB 4
00018 EXITSTATUS: .BLKB 4
00000000 0001C GETQUIEXITBLOCK: .BLKB 4
00000000V 00020 .LONG 0
00000001 00024 .ADDRESS CANCEL_GETQUI
00000000' 00028 .LONG 1
00000000' 0002C EXITBLOCK: .ADDRESS EXITSTATUS
00000000V 00030 .LONG 0
00000001 00034 .ADDRESS CLOSE_JOB
00000000' 00038 .LONG 1
0003C EXITDECLARED: .ADDRESS EXITSTATUS
00000000' 0003C EXITDECLARED: .BLKB 4

FF_STRING= P.AAA
LISTFILE_DESC= P.AAB
SD_APPEND= P.AAD
SD_ALL= P.AAF
SD_EXTF= P.AAH
SD_HEADER= P.AAJ
SD_MAIL= P.AAL

.EXTRN MAIL\$GL_FLAGS, MAIL\$L_NMSG
.EXTRN MAIL\$L_TURMSG, MAIL\$L_MSG_NUM
.EXTRN MAIL\$L_MSG_STS, MAIL\$L_MSG_RADR
.EXTRN MAIL\$W_MSG_RSIZ
.EXTRN MAIL\$L_MSGBUFSIZ
.EXTRN MAIL\$Q_USERDEF, MAIL\$B_MSGTRAN
.EXTRN CLISPRESENT, CLISGET VALUE
.EXTRN MAIL\$FIND_MSG, MAIL\$FIND_I_MSG
.EXTRN MAIL\$GET_I_MSG, MAIL\$GET_MSG
.EXTRN MAIL\$GET_MSG, MAIL\$PUT_OUTPUT
.EXTRN MAIL\$REWIND_I_MSG
.EXTRN MAIL\$REWIND_MSG
.EXTRN MAIL\$TEST_SOBJ, STR\$COPY_DX
.EXTRN STR\$CONCAT, STR\$FREE1_DX
.EXTRN UTIL\$GETFILENAME
.EXTRN UTIL\$REPORT_IO_ERROR
.EXTRN SY\$PUT

.PSECT \$CODE\$, NOWRT, 2

0000 00000 WRITE_REC:
00000000G 00 9F 00002 .WORD Save nothing
04 AC DD 00008 PUSHAB UTIL\$REPORT_IO_ERROR
00000000G 00 02 FB 0000B PUSHBL SND_RAB
04 00012 CALLS #2, SY\$PUT
RET

: Routine Size: 19 bytes, Routine Base: \$CODE\$ + 0064

: 0320
: 0332
: 0333

```

192 0334 1 ROUTINE WRITE_FF (SND_RAB) =
193 0335 1 ++
194 0336 1 FUNCTIONAL DESCRIPTION:
195 0337 1
196 0338 1 Write formfeed record to output file
197 0339 1
198 0340 1 INPUTS:
199 0341 1
200 0342 1 SND_RAB is address of rab
201 0343 1 --
202 0344 2 BEGIN
203 0345 2 MAP
204 0346 2 SND_RAB : REF $BBLOCK;
205 0347 2
206 0348 2 SND_RAB[RAB$W_RSZ] = 1;
207 0349 2 SND_RAB[RAB$L_RBF] = FF_STRING;
208 0350 2
209 0351 2 RETURN WRITE_REC(.SND_RAB)
210 0352 1 END;

```

0004 00000 WRITE_FF:

					WORD	Save R2	: 0334
	52	84	AF	9E 00002	MOVAB	FF STRING, R2	
	50	04	AC	D0 00006	MOVL	SND_RAB, R0	: 0348
22	A0		01	B0 0000A	MOVW	#1,-34(R0)	
28	A0		62	9E 0000E	MOVAB	FF_STRING, 40(R0)	: 0349
			50	DD 00012	PUSHL	R0	: 0351
D5	AF		01	FB 00014	CALLS	#1, WRITE_REC	
			04	00018	RET		: 0352

; Routine Size: 25 bytes, Routine Base: \$CODE\$ + 0077

```

212 0353 1 GLOBAL ROUTINE MAIL$EXTRACT_MESSAGE(HEADER_FLAG,SND_RAB) =
213 0354 1 ++
214 0355 1 FUNCTIONAL DESCRIPTION:
215 0356 1
216 0357 1 Extract the current msg to the output file
217 0358 1
218 0359 1 INPUTS:
219 0360 1
220 0361 1 HEADER_FLAG = 1 to include msg header, 0 if not
221 0362 1 SND_RAB is address of output rab
222 0363 1 --
223 0364 2 BEGIN
224 0365 2
225 0366 2 MAP
226 0367 2 SND_RAB : REF $BBBLOCK;
227 0368 2
228 0369 2 LOCAL
229 0370 2 MORE_FLAG,
230 0371 2 DESC : VECTOR[2,LONG];
231 0372 2
232 0373 2 IF .MAIL$GL_FLAGS[MAIF_V_ISAM]
233 0374 3 THEN BEGIN
234 0375 3 MAIL$REWIND I MSG();
235 0376 3 IF NOT .HEADER_FLAG
236 0377 3 THEN MAIL$B MSGTRAN = MMB C TEXT;
237 0378 3 WHILE MAIL$GET_I_MSG(DESC,MORE_FLAG)
238 0379 4 DO BEGIN
239 0380 4 SND_RAB[RABSW_RSZ] = .DESC[0];
240 0381 4 SND_RAB[RABSL_RBF] = .DESC[1];
241 0382 4 P IF_ERR(WRITE REC(.SND_RAB));
242 0383 4 RETURN .STATUS;
243 0384 3
244 0385 3 END;
245 0386 3
246 0387 3 ELSE BEGIN
247 0388 3 MAIL$REWIND MSG();
248 0389 4 IF NOT .HEADER_FLAG
249 0390 4 THEN INCRU_I FROM 0 TO (IF MAIL$TEST_SUBJ() THEN 3 ELSE 2)
250 0391 3 DO IF NOT MAIL$GET_MSG() THEN EXITLOOP;
251 0392 4
252 0393 4 DO BEGIN
253 0394 4 SND_RAB[RABSW_RSZ] = .MAIL$W_MSG_RSIZ;
254 0395 4 SND_RAB[RABSL_RBF] = .MAIL$L_MSG_RADR;
255 0396 4 P IF_ERR(WRITE REC(.SND_RAB));
256 0397 4 RETURN .STATUS;
257 0398 3
258 0399 2
259 0400 2 RETURN TRUE
260 0401 1 END;

```

00FC 00000
57 00000000G 00 9E 00002

.ENTRY MAIL\$EXTRACT_MESSAGE, Save R2,R3,R4,R5,R6,- : 0353
R7
MOVAB MAIL\$GET_MSG, R7

MAIL\$PRINT
VO4-000

QUEUE MAIL MESSAGE FOR PRINTING

D 2
16-Sep-1984 01:14:12
14-Sep-1984 12:42:29

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]PRINT.B32;1

Page 9
(5)

56	C8	AF	9E	00009	MOVAB	WRITE REC, R6			
5E	0C	C2	0000D	SUBL2	#12, SP	0376			
52	04	AC	D2	00010	MCOML	HEADER FLAG, R2	0380		
55	08	AC	D0	00014	MOVL	SND_RAB, R5	0381		
54	08	AC	D0	00018	MOVL	SND_RAB, R4	0381		
34	00000000G	00	E9	0001C	BLBC	MAI\$GL_FLAGS, 2\$	0373		
00000000G	00	00	FB	00023	CALLS	#0, MAI\$REWIND_I_MSG	0375		
07	52	E9	0002A	BLBC	R2, 1\$	0376			
00000000G	00	05	90	0002D	MOVB	#5, MAIL\$B_MSGTRAN	0377		
	5E	DD	00034	1\$:	PUSHL	SP	0378		
00000000G	00	08	AE	9F	PUSHAB	DESC			
	02	FB	00039		CALLS	#2, MAIL\$GET_I_MSG			
22	A5	04	AE	B0	BLBC	R0, 8\$	0380		
28	A4	08	AE	D0	MOVW	DESC, 34(R5)	0381		
	08	AC	DD	00048	MOVL	DESC+4, 40(R4)	0381		
66	01	FB	00050	PUSHL	SND_RAB	0383			
DE	50	E8	00053	CALLS	#1, WRITE REC				
	04	00056		BLBS	STATUS, 1\$				
RET									
00000000G	00	00	FB	00057	2\$:	CALLS	#0, MAIL\$REWIND_MSG	0387	
23	52	E9	0005E	BLBC	R2, 7\$	0388			
00000000G	00	00	FB	00061	CALLS	#0, MAIL\$TEST_SUBJ	0389		
05	50	E9	00068	BLBC	R0, 3\$				
52	03	D0	0006B	MOVL	#3, R2				
	03	11	0006E	BRB	4\$				
52	02	D0	00070	3\$:	MOVL	#2, R2			
	53	D4	00073	4\$:	CLRL	I			
	08	11	00075		BRB	6\$			
67	00	FB	00077	5\$:	CALLS	#0, MAIL\$GET_MSG	0390		
07	50	E9	0007A	BLBC	R0, 7\$				
	53	D6	0007D	INCL	I				
52	53	D1	0007F	6\$:	CMPL	I, R2			
	F3	1B	00082	BLEQU	5\$				
22	A5	00000000G	00	B0	00084	7\$:	MOVW	MAIL\$W_MSG_RSIZE, 34(R5)	0393
28	A4	00000000G	00	D0	0008C		MOVL	MAIL\$L_MSG_RADDR, 40(R4)	0394
	08	AC	DD	00094	PUSHL	SND_RAB	0396		
66	01	FB	00097	CALLS	#1, WRITE REC				
09	50	E9	0009A	BLBC	STATUS, 9\$				
67	00	FB	0009D	CALLS	#0, MAIL\$GET_MSG	0398			
E1	50	E8	000A0	BLBS	R0, 7\$				
50	01	D0	000A3	8\$:	MOVL	#1, R0			
	04	000A6	9\$:		RET				
							0400		
							0401		

; Routine Size: 167 bytes, Routine Base: \$CODE\$ + 0090

```

262 0402 1 ROUTINE EXTRACT_ALL_ISAM(HEADER_FLAG,SND_RAB) =
263 0403 1 ++
264 0404 1 FUNCTIONAL DESCRIPTION:
265 0405 1
266 0406 1 Extract all messages from the current folder/seq. file to the output
267 0407 1 file.
268 0408 1
269 0409 1 INPUTS:
270 0410 1
271 0411 1 HEADER_FLAG = 1 if all msgs, 0 if current msg only
272 0412 1 SND_RAB is address of output file rab
273 0413 1
274 0414 1 --
275 0415 2 BEGIN
276 0416 2
277 0417 2 LOCAL
278 0418 2 STATUS;
279 0419 2
280 0420 2 BUILTIN
281 0421 2 AP,
282 0422 2 CALLG;
283 0423 2
284 0424 2 MAIL$L_MSG_NUM = 0;
285 0425 2 WHILE MAIL$FIND_I_MSG(0)
286 0426 3 DO BEGIN
287 0427 3 IF .MAIL$L_MSG_NUM NEQ 1
288 0428 3 THEN WRITE_FF(.SND_RAB);
289 0429 4 IF NOT (STATUS = CALLG7.AP,MAIL$EXTRACT_MESSAGE))
290 0430 3 THEN RETURN .STATUS;
291 0431 2 END;
292 0432 2 RETURN TRUE
293 0433 1 END;

```

000C 00000 EXTRACT_ALL_ISAM:

				WORD	Save R2,R3	0402
				MOVAB	MAIL\$L_MSG_NUM, R3	0424
				CLRL	MAIL\$L_MSG_NUM	0425
				CLRL	-(SP)	0426
				CALLS	#1, MAIL\$FIND_I_MSG	0427
				BLBC	R0, 3\$	0428
				CMPL	MAIL\$L_MSG_NUM, #1	0429
				BEQL	2\$	0430
				PUSHL	SND_RAB	0431
				CALLS	#1, WRITE_FF	0432
				CALLG	(AP), MAIL\$EXTRACT_MESSAGE	0433
				MOVL	R0, STATUS	
				BLBS	R0, 1\$	
				MOVL	STATUS, R0	
				RET		
				MOVL	#1, R0	
				RET		

; Routine Size: 55 bytes, Routine Base: \$CODE\$ + 0137

MAIL\$PRINT
V04-000

QUEUE MAIL MESSAGE FOR PRINTING

F 2
16-Sep-1984 01:14:12 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:42:29 DISK\$VMSMASTER:[MAIL.SRC]PRINT.B32;1 Page 11
(6)

```

295 0434 1 ROUTINE EXTRACT_ALL_SEQ(HEADER_FLAG,SND_RAB) =
296 0435 1 ++
297 0436 1 FUNCTIONAL DESCRIPTION:
298 0437 1
299 0438 1 Extract all sequential msgs to the output file
300 0439 1
301 0440 1 INPUTS:
302 0441 1
303 0442 1 HEADER_FLAG = 1 to include msg headers, 0 if not
304 0443 1 SND_RAB is address of output rab
305 0444 1
306 0445 1 !--
307 0446 2 BEGIN
308 0447 2
309 0448 2 BUILTIN
310 0449 2 AP,
311 0450 2 CALLG;
312 0451 2
313 0452 2 LOCAL
314 0453 2 STATUS;
315 0454 2 MAIL$L_MSG_NUM = 0;
316 0455 2 WHILE MAIL$FIND_MSG(0)
317 0456 3 DO BEGIN
318 0457 3 IF .MAIL$L_MSG_NUM NEQ 1
319 0458 3 THEN WRITE-FF(.SND_RAB);
320 0459 4 IF NOT (STATUS = CALLG7.AP,MAIL$EXTRACT_MESSAGE)
321 0460 3 THEN RETURN .STATUS;
322 0461 2 END;
323 0462 2 RETURN TRUE
324 0463 1 END;

```

000C 00000 EXTRACT_ALL SEQ:

				WORD	Save R2,R3	0434
				MOVAB	MAIL\$L_MSG_NUM, R3	0454
				CLRL	MAIL\$L_MSG_NUM	0455
				CLRL	-(SP)	
				CALLS	#1, MAIL\$FIND_MSG	0457
				BLBC	R0, 3\$	
				CMPL	MAIL\$L_MSG_NUM, #1	
				BEQL	2\$	
				PUSHL	SND_RAB	0458
				CALLS	#1, -WRITE_FF	
				CALLG	(AP), MAIL\$EXTRACT_MESSAGE	0459
				MOVL	R0, STATUS	
				BLBS	R0, 1\$	
				MOVL	STATUS, R0	0460
				RET		
				MOVL	#1, R0	0462
				RET		0463

: Routine Size: 55 bytes. Routine Base: \$CODE\$ + 016E

```
326 0464 1 GLOBAL ROUTINE MAIL$CMD_EXTRACT =
327 0465 1 ++
328 0466 1 FUNCTIONAL DESCRIPTION:
329 0467 1
330 0468 1 Handle the EXTRACT command, to extract a message out to a separate
331 0469 1 sequential file
332 0470 1 --
333 0471 2 BEGIN
334 0472 2
335 0473 2 LOCAL
336 0474 2 SND_FAB : $FAB_DECL,
337 0475 2 SND_RAB : $RAB_DECL,
338 0476 2 SND_NAM : $NAM_DECL,
339 0477 2 SND_RSS : $BBLLOCK[NAMSC MAXRSS],
340 0478 2 SND_XABPRO : $XABPRO DECL,
341 0479 2 SND_DESC : VECTOR[2,[ONG];
342 0480 2 CREATE_STATUS,
343 0481 2 STATUS,
344 0482 2 HEAD_FLAG : BYTE,
345 0483 2 APPEND_FLAG : BYTE,
346 0484 2 MAIL_FLAG : BYTE,
347 0485 2 ALL_FLAG : BYTE;
348 0486 2
349 0487 2 OWN
350 0488 2 DYN_DESC : DYN_DESC_DECL;
351 0489 2
352 0490 2 ALL_FLAG = CLI$PRESENT(SD_ALL);
353 0491 2 APPEND_FLAG = CLI$PRESENT(SD_APPEND);
354 0492 2 HEAD_FLAG = CLI$PRESENT(SD_HEADER);
355 0493 2 MAIL_FLAG = CLI$PRESENT(SD_MAIL);
356 0494 2 IF .MAIL_FLAG
357 0495 2 AND NOT .HEAD_FLAG
358 0496 2 THEN RETURN SIGNAL(MAIL$_CONFQUAL);
359 0497 2
360 P 0498 2 IF _ERR(CLIGET_VALUE(SD_EXTF,DYN_DESC));
361 0499 2 RETURN STATUS;
362 0500 2
363 0501 2 IF NOT .ALL_FLAG
364 0502 3 THEN BEGIN
365 0503 4 IF (IF .MAIL$GL_FLAGS[MAIF V ISAM]
366 0504 4 THEN .MAIL$[ CURMSG EQ[ 0
367 0505 4 ELSE (NOT .MAIL$L_MSG_STS AND .MAIL$L_MSG_STS NEQ 0))
368 0506 3 THEN RETURN SIGNAL(MAIL$_NOTREADIN);
369 0507 3 END
370 0508 3 ELSE BEGIN
371 0509 4 IF (IF .MAIL$GL_FLAGS[MAIF V_ISAM]
372 0510 4 THEN .MAIL$[ NMSG EQ[ 0
373 0511 4 ELSE .MAIL$L_MSGBUFSIZ EQ[ 0)
374 0512 3 THEN RETURN SIGNAL(MAIL$_NOMSGS);
375 0513 2 END;
376 0514 2
377 P 0515 2 $FAB_INIT(FAB=SND_FAB,
378 P 0516 2 FNS=.DYN_DESC[DSC$W_LENGTH],
379 P 0517 2 FNA=.DYN_DESC[DSC$A_POINTER],
380 P 0518 2 DNM=.TXT,
381 P 0519 2 FAC=(PUT TRN),
382 P 0520 2 CTX=MAIL$_OPENOUT,
```

```
383 P 0521 2 NAM=SND_NAM,  
384 P 0522 2 RAT=(R,  
385 P 0523 2 RFM=VAR);  
386 P 0524 2  
387 P 0525 2 $XABPRO_INIT(XAB=SND_XABPRO,  
388 P 0526 2 UIC=(0,0));  
389 P 0527 2  
390 P 0528 2 IF .MAIL_FLAG  
391 P 0529 3 THEN BEGIN  
392 P 0530 3 APPEND_FLAG = 1;  
393 P 0531 3 IF .DEFDIRDESC[DSC$W_LENGTH] EQL 0  
394 P 0532 3 THEN STR$CONCAT(DEFDIRDESC,MAIL$Q_USERDEF,$DESCRIPTOR('.MAI'));  
395 P 0533 3 SND_FAB[FAB$B_DNS] = .DEFDIRDESC[DSC$W_LENGTH];  
396 P 0534 3 SND_FAB[FAB$L_DNA] = .DEFDIRDESC[DSC$A_POINTER];  
397 P 0535 3 SND_FAB[FAB$L_XAB] = SND_XABPRO;  
398 P 0536 3 SND_XABPRO[XAB$W_PRO] = SND_PRO;  
399 P 0537 2 END;  
400 P 0538 2  
401 P 0539 2 $NAM_INIT(NAM=SND_NAM,  
402 P 0540 2 ESA=SND_RSS,  
403 P 0541 2 ESS=NAM$C_MAXRSS,  
404 P 0542 2 RSA=SND_RSS,  
405 P 0543 2 RSS=NAM$C_MAXRSS);  
406 P 0544 2 SND_DESC[1] = SND_RSS;  
407 P 0545 2  
408 P 0546 2 IF .APPEND_FLAG  
409 P 0547 2 THEN SND_FAB[FAB$V_CIF] = TRUE;  
410 P 0548 2  
411 P 0549 2 CREATE_STATUS = $CREATE(FAB=SND_FAB,ERR=UTIL$REPORT_IO_ERROR);  
412 P 0550 2 IF NOT .CREATE_STATUS  
413 P 0551 2 THEN RETURN .CREATE_STATUS;  
414 P 0552 2  
415 P 0553 2 IF .SND_FAB[FAB$B_ORG] NEQ FAB$C_SEQ  
416 P 0554 3 THEN BEGIN  
417 P 0555 3 SND_FAB[FAB$L_STS] = RMSS_IOP;  
418 P 0556 3 UTI[$REPORT_IO_ERROR(SND_FAB);  
419 P 0557 3 SCLOSE(FAB=SND_FAB);  
420 P 0558 3 RETURN RMSS_IOP  
421 P 0559 2 END;  
422 P 0560 2  
423 P 0561 2 $RAB_INIT(RAB=SND_RAB,  
424 P 0562 2 CTX=MAIL$OPENOUT,  
425 P 0563 2 FAB=SND_FAB);  
426 P 0564 2  
427 P 0565 2 IF .APPEND_FLAG  
428 P 0566 2 THEN SND_RAB[RAB$V_EOF] = TRUE;  
429 P 0567 2  
430 P 0568 2 IF _ERR($CONNECT(RAB=SND_RAB,ERR=UTIL$REPORT_IO_ERROR));,  
431 P 0569 2 IF NOT .APPEND_FLAG  
432 P 0570 2 THEN SND_FAB[FAB$V_DLT] = 1;  
433 P 0571 2 SCLOSE(FAB=SND_FAB);  
434 P 0572 2 RETURN .STATUS;  
435 P 0573 2  
436 P 0574 2 SND_RAB[RAB$L_CTX] = MAIL$WRITEERR;  
437 P 0575 2 CHSMOVE(DSC$C_S_BLN,UTIL$GETFILENAME(SND_FAB),SND_DESC);  
438 P 0576 2  
439 P 0577 3 IF (IF .APPEND_FLAG
```

```

440 0578 4      THEN (.CREATE STATUS EQL RMSS$ CREATED)
441 0579 3      ELSE NOT .SND_FAB[$FAB DEV(TRM)])
442 0580 2      THEN IF .MAIL$GL FLAGS[MAIF V ITERM]
443 0581 2      THEN SIGNAL(MAIL$_CREATED,1,SND_DESC);
444 0582 2
445 0583 2      IF .APPEND_FLAG
446 0584 2      THEN WRITE_FF(SND_RAB);
447 0585 2
448 0586 2      IF NOT .ALL_FLAG
449 0587 2      THEN STATUS= MAIL$EXTRACT MESSAGE(.HEAD_FLAG,SND_RAB)
450 0588 2      ELSE IF .MAIL$GL FLAGS[MAIF V_ISAM]
451 0589 2      THEN STATUS= EXTRACT_ALL_ISAM(.HEAD_FLAG,SND_RAB)
452 0590 2      ELSE STATUS= EXTRACT_ALL_SEQ(.HEAD_FLAG,SND_RAB);
453 0591 2
454 0592 2      SND_FAB[FAB$L_CTX] = MAIL$ CLOSEOUT;
455 0593 2      $CLOSE(FAB=SND_FAB,ERR=UTI[$REPORT_IO_ERROR]);
456 0594 2
457 0595 2      RETURN .STATUS
458 0596 1      END;

```

```

54 58 54 2E 001A5 P.AAN: .ASCII \.TXT\
49 41 4D 2E 001A9 P.AAP: .ASCII \.MAI\
001AD .BLKB 3
00000004 001B0 P.AAO: .LONG 4
00000000 001B4 .ADDRESS P.AAP
.PSECT $OWNS,NOEXE,2

```

```

00# 00040 DYN_DESC:
02 0E 00042 .BYTE 0[2]
00044 .BYTE 14, 2
00044 .BLKB 4
.EXTRN SYSSCREATE, SYSSCLOSE
.EXTRN SYSSCONNECT
.PSECT $CODE$,NOWRT,2

```

		OFFC 00000	.ENTRY	0464
5B	00000000	00 9E 00002	MOVAB DEFDIRDESC, R11	
5A	FE6F	CF 9E 00009	MOVAB SD_ALL, R10	
5E	FDAC	CE 9E 0000E	MOVAB -596(SP), SP	
00000000G	00	5A DD 00013	PUSHL R10	
59		01 FB 00015	CALLS #1, CLISPRES	0490
00000000G	00	50 90 0001C	MOVBL R0, ALL_FLAG	
57	F4	AA 9F 0001F	PUSHAB SD_APPEND	0491
00000000G	00	01 FB 00022	CALLS #1, CLISPRES	
57		50 90 00029	MOVBL R0, APPEND_FLAG	
00000000G	00	1C AA 9F 0002C	PUSHAB SD_HEADER	0492
58		01 FB 00036	CALLS #1, CLISPRES	
00000000G	00	50 90 00039	MOVBL R0, HEAD_FLAG	
56	28	AA 9F 0003C	PUSHAB SD_MAIL	0493
		01 FB 00043	CALLS #1, CLISPRES	
		50 90 00043	MOVBL R0, MAIL_FLAG	

MAIL\$PRINT
VO4-000

QUEUE MAIL MESSAGE FOR PRINTING

K 2
16-Sep-1984 01:14:12
14-Sep-1984 12:42:29 VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]PRINT.B32;1Page 16
(8)

08	08	0000000G 00	007E12E0	56 E9 00046	BLBC	MAIL_FLAG, 1\$	0494
08	08			58 E8 00049	BLBS	HEAD_FLAG, 1\$	0495
				8F DD 0004C	PUSHL	#8262368	0496
				57 11 00052	BRB	8\$	
				38 AB 9F 00054	1\$: PUSHAB	DYN_DESC	0499
				0C AA 9F 00057	PUSHAB	SD_EXTF	
				00 02 FB 0005A	CALLS	#2_CLI\$GET_VALUE	
				01 50 E8 00061	BLBS	STATUS, 2\$	
				04 04 00064	RET		
50	0000000G 00			01 00 EF 00065	EXTZV	#0, #1_MAIL\$GL_FLAGS, R0	0503
				21 59 E8 0006E	BLBS	ALL_FLAG, 5\$	0501
				0A 50 E9 00071	BLBC	R0_3\$	0503
			0000000G	00 D5 00074	TSTL	MAIL\$L_CURMSG	0504
				37 12 0007A	BNEQ	9\$	
			0000000G	0C 11 0007C	BRB	4\$	
				50 0000000G 00 D0 0007E	3\$: MOVL	MAIL\$L_MSG_STS, R0	0505
				2B 50 E8 00085	BLBS	R0, 9\$	
			0000000G	29 13 00088	BEQL	9\$	
				08 19 11 00090	PUSHL	#8290338	0506
			0000000G	50 E9 00092	BRB	8\$	
				00 D5 00095	BLBC	R0, 6\$	0509
			0000000G	06 11 0009B	TSTL	MAIL\$L_NMSG	0510
				00 D5 0009D	BNEQ	7\$	
			0000000G	0E 12 000A3	TSTL	MAIL\$L_MSGBUFSIZ	0511
				7\$: 007E804A 8F DD 000A5	BNEQ	9\$	
			0000000G	01 FB 000AB	PUSHL	#8290378	0512
				04 000B2	CALLS	#1_LIB\$SIGNAL	
0050	8F	00	6E	00 2C 000B3	RET		
				9\$: B0 AD 5003	MOVCS	#0, (SP), #0, #80, \$RMS_PTR	0523
				8F B0 000BA	MOVW	#20483, \$RMS_PTR	
				C6 AD 11 90 000C2	MOVB	#17 \$RMS_PTR+22	
				C8 AD 007E10A2 8F D0 000C6	MOVL	#8261794, \$RMS_PTR+24	
				CE AD 0202 8F B0 000CE	MOVW	#514, \$RMS_PTR+30	
				D8 AD FFOC CD 9E 000D4	MOVAB	SND_NAM, \$RMS_PTR+40	
				DC AD 3C AB D0 000DA	MOVL	DYN_DESC+4, \$RMS_PTR+44	
				E0 AD 0171 CA 9E 000DF	MOVAB	P.AAN, \$RMS_PTR+48	
				E4 AD 38 AB 90 000E5	MOVB	DYN_DESC, \$RMS_PTR+52	
0058	8F	00	E5	04 90 000EA	MOVCS	#4 \$RMS_PTR+53	
			6E	00 2C 000EE	MOVW	#0, (SP), #0, #88, \$RMS_PTR	0526
				AE 000F5	MOVB	#22547, \$RMS_PTR	
			08 AE 5813	8F B0 000F7	BLBC	MAIL_FLAG, 1T\$	0528
				56 E9 000FD	MOVBL	#1_APPEND_FLAG	0530
				2C 01 90 00100	TSTW	DEFDIRDESC	0531
				57 6B B5 00103	BNEQ	10\$	
				017C 13 12 00105	PUSHAB	P.AAO	0532
			0000000G	CA 9F 00107	PUSHAB	MAIL\$Q_USERDEF	
				00 9F 0010B	PUSHL	R11	
			0000000G 00	5B DD 00111	CALLS	#3_STR\$CONCAT	
			E5 AD 03 FB 00113	MOVBL	DEFDIRDESC, SND_FAB+53	0533	
			E0 AD 6B 90 0011A	MOVBL	DEFDIRDESC+4, SND_FAB+48	0534	
			D4 AD 04 AB D0 0011E	MOVAB	SND_XABPRO, SND_FAB+36	0535	
			10 AE 08 AE 9E 00123	MNEGW	#52 SND_XABPRO+8	0536	
0060	8F	00	6E	34 AE 00128	MOVCS	#0, (SP), #0, #96, \$RMS_PTR	0543
			FFOC CD 00 2C 0012C	11\$: 6002 8F B0 00136	MOVW	#24578, \$RMS_PTR	

FF0E	CD	01	8E	0013D	MNEG B	#1, \$RMS_PTR+2		
FF10	CD	60	AE	00142	MOVAB	SN D_RSS -\$RMS_PTR+4		
FF16	CD	01	8E	00148	MNEG B	#1, \$RMS_PTR+10		
FF18	CD	60	AE	0014D	MOVAB	SN D_RSS -\$RMS_PTR+12		
04	AE	60	AE	00153	MOVAB	SN D_RSS -\$DESC+4	0544	
04	04	57	E9	00158	BLBC	APPEND FLAG, 12\$	0546	
B7	AD	02	88	0015B	BISB2	#2 SND_FAB+7	0547	
		00	9F	0015F	12\$:	PUSHAB	UTIL\$REPORT_IO_ERROR	0549
		B0	AD	00165	PUSHAB	SND_FAB		
00000000G	00	02	FB	00168	CALLS	#2, SYSSCREATE		
56		50	DO	0016F	MOVL	RO, CREATE STATUS		
04		56	E8	00172	BLBS	CREATE_STATUS, 13\$	0550	
50		56	DO	00175	MOVL	CREATE_STATUS, RO	0551	
		04		00178	RET			
		CD	AD	95 00179	13\$:	TSTB	SND_FAB+29	0553
		24	13	0017C	BEQL	14\$-		
B8	AD 00018574	8F	DO	0017E	MOVL	#99700, SND_FAB+8	0555	
		B0	AD	9F 00186	PUSHAB	SND_FAB	0556	
00000000G	00	01	FB	00189	CALLS	#1, UTIL\$REPORT_IO_ERROR		
		B0	AD	9F 00190	PUSHAB	SND_FAB	0557	
00000000G	00	01	FB	00193	CALLS	#1, SYSSCLOSE		
		50 00018574	8F	DO 0019A	MOVL	#99700, RO	0558	
		04		001A1	RET			
0044 8F	00	6E	00	2C 001A2	14\$:	MOVCS	#0, (SP), #0, #68, \$RMS_PTR	0563
		FF6C	CD	001A9				
FF6C	CD	4401	8F	B0 001AC	MOVW	#17409, \$RMS_PTR		
84	AD 007E10A2	8F	DO	001B3	MOVL	#8261794, \$RMS_PTR+24		
A8	AD	B0	AD	9E 001BB	MOVAB	SND_FAB, \$RMS_PTR+60		
05		57	E9	001C0	BLBC	APPEND FLAG, T5\$	0565	
FF71	CD	01	88	001C3	BISB2	#1, SND_FAB+5	0566	
		00000000G	00	9F 001C8	15\$:	PUSHAB	UTIL\$REPORT_IO_ERROR	0572
		FF6C	CD	9F 001CE	PUSHAB	SND_FAB		
00000000G	00	02	FB	001D2	CALLS	#2, SYSSCONNECT		
52		50	DO	001D9	MOVL	RO, STATUS		
15		52	E8	001DC	BLBS	STATUS, 17\$		
05		57	E8	001DF	BLBS	APPEND FLAG, 16\$		
B5	AD	80	8F	88 001E2	BISB2	#128, SND_FAB+5		
		B0	AD	9F 001E7	16\$:	PUSHAB	SND_FAB	
00000000G	00	01	FB	001EA	CALLS	#1, SYSSCLOSE		
		009B	31	001F1	BRW	25\$		
84	AD 007E10D2	8F	DO	001F4	17\$:	MOVL	#8261842, SND_FAB+24	0574
		B0	AD	9F 001FC	PUSHAB	SND_FAB	0575	
6E	00000000G	00	01	FB 001FF	CALLS	#1, UTIL\$GETFILENAME		
60		08	28	00206	MOVC3	#8, (RO), SND_DESC		
0B		57	E9	0020A	BLBC	APPEND FLAG, T8\$	0577	
00010619	8F	56	D1	0020D	CMPL	CREATE_STATUS, #67097	0578	
		20	12	00214	BNEQ	20\$		
19		05	11	00216	BRB	19\$		
11 00000000G	00	02	E0	00218	18\$:	BBS	#2, SND_FAB+64, 20\$	0579
		02	E1	0021D	19\$:	BBC	#2, MAIL\$GL_FLAGS, 20\$	0580
		5E	DD	00225	PUSHL	SP	0581	
		01	DD	00227	PUSHL	#1		
00000000G	00	007E1073	8F	DD 00229	PUSHL	#8261747		
09		03	FB	0022F	CALLS	#3, LIB\$SIGNAL		
		57	E9	00236	BLBC	APPEND FLAG, 21\$	0583	
FC7D	CF	FF6C	CD	9F 00239	PUSHAB	SND_FAB	0584	
		01	FB	0023D	CALLS	#1, WRITE_FF		

MAIL\$PRINT
VO4-000

QUEUE MAIL MESSAGE FOR PRINTING

M 2
16-Sep-1984 01:14:12
14-Sep-1984 12:42:29
VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]PRINT.B32;1

Page 18
(8)

	OE	FF6C	59	E8	00242	21\$:	BLBS	ALL_FLAG, 22\$	0587
	7E		CD	9F	00245		PUSHAB	SND_RAB	
FC87	CF		58	9A	00249		MOVZBL	HEAD_FLAG, -(SP)	
			02	FB	0024C		CALLS	#2 MAIL\$EXTRACT_MESSAGE	
			21	11	00251		BRB	24\$	
	OE	00000000G	00	E9	00253	22\$:	BLBC	MAIL\$GL_FLAGS, 23\$	0588
		FF6C	CD	9F	0025A		PUSHAB	SND_RAB	0589
FD19	7E		58	9A	0025E		MOVZBL	HEAD_FLAG, -(SP)	
	CF		02	FB	00261		CALLS	#2 EXTRACT_ALL_ISAM	
			0C	11	00266		BRB	24\$	
		FF6C	CD	9F	00268	23\$:	PUSHAB	SND_RAB	0590
FD42	7E		58	9A	0026C		MOVZBL	HEAD_FLAG, -(SP)	
	CF		02	FB	0026F		CALLS	#2, EXTRACT_ALL_SEQ	
	52		50	DO	00274	24\$:	MOVL	RO, STATUS	
C8	AD	007E105A	8F	DO	00277		MOVL	#8261722, SND_FAB+24	0592
		00000000G	00	9F	0027F		PUSHAB	UTIL\$REPORT_10_ERROR	0593
		BO	AD	9F	00285		PUSHAB	SND_FAB	
00000000G	00		02	FB	00288		CALLS	#2, SYS\$CLOSE	
	50		52	DO	0028F	25\$:	MOVL	STATUS, RO	0595
			04	00292			RET		0596

; Routine Size: 659 bytes, Routine Base: \$CODE\$ + 01B8

```
; 460      0597 1 ROUTINE CANCEL_GETQUI : NOVALUE =
; 461      0598 2 BEGIN
; 462      0599 2 ++
; 463      0600 2 This routine cancels any GETQUI in progress. Called if an image exit done
; 464      0601 2 while getqui request outstanding
; 465      0602 2 --
; 466      0603 2 --
; 467      0604 2
; 468      0605 2 $GETQUIW(FUNC=QUI$CANCEL_OPERATION);
; 469      0606 2 RETURN;
; 470      0607 1 END;
```

.EXTRN SYSS\$GETQUIW

	0000 00000 CANCEL_GETQUI:		
	7E 7C 00002	WORD	Save nothing
	7E 7C 00004	CLRQ	-(SP)
7E	01 7D 00006	CLRQ	-(SP)
	7E D4 00009	MOVQ	#1, -(SP)
0000000G 00	07 FB 0000B	CLRL	-(SP)
	04 00012	CALLS	#7, SYSS\$GETQUIW
		RET	

; 0597
; 0605
; 0607

; Routine Size: 19 bytes, Routine Base: \$CODE\$ + 044B

```
472      0608 1 ROUTINE GET_FLAG_TRAILER (FLAG_ITEM,TRAILER_ITEM) : NOVALUE =
473      0609 2 BEGIN
474      0610 2 ++
475      0611 2 This routine does a $GETQUIW on the current queue (in CURQUE_DESC)
476      0612 2 and returns the correct flag and trailer options to use in create job.
477      0613 2
478      0614 2 IMPLICIT INPUTS:
479      0615 2
480      0616 2      curque_desc = descriptor of queue
481      0617 2
482      0618 2
483      0619 2      FLAG_ITEM contains the sjc$_ code for flg page, TRAILER_ITEM
484      0620 2      contains code for trailer page.
485      0621 2
486      0622 2 --
487      0623 2 LOCAL
488      0624 2      GETQLST : $ITMLST DECL(ITEMS=3),
489      0625 2      QUEUE_FLAGS : $BB[LOCK[8] VOLATILE,
490      0626 2      IOSB : VECTOR[4,WORD],
491      0627 2      SEARCH_FLAGS,
492      0628 2      STATUS;
493      0629 2
494      0630 2 .FLAG ITEM = SJCS FILE FLAG ONE;
495      0631 2 .TRAILER ITEM = SJCS_NO FILE TRAILER;
496      0632 2 SEARCH_FLAGS = QUI$M_SEARCH_WILDCARD OR QUI$M_SEARCH_SYMBIONT;
497      P 0633 2 $ITMLST_INIT(ITMLST=GETQLST,
498      P 0634 2      (ITMCOD=QUI$ SEARCH NAME,
499      P 0635 2          BUFADR=.CURQUE_DESC[DSC$A_POINTER],
500      P 0636 2          BUFSIZE=.CURQUE_DESC[DSC$W_LENGTH]),
501      P 0637 2      (ITMCOD=QUI$ SEARCH FLAGS,
502      P 0638 2          BUFADR=SEARCH_FLAGS,
503      P 0639 2          BUFSIZ=4),
504      P 0640 2      (ITMCOD=QUI$_QUEUE_FLAGS,BUFADR=QUEUE_FLAGS));
505      0641 2
506      0642 2 $GETQUIW(FUNC=QUI$ CANCEL OPERATION);
507      0643 2 $DCLEXH(DESBLK=GETQUIEXITBLOCK);
508      P 0644 2 STATUS = $GETQUIW(FUNC=QUI$ DISPLAY_QUEUE,
509      P 0645 2      ITMLST=GETQLST,
510      0646 2      IOSB=IOSB);
511      0647 2 $GETQUIW(FUNC=QUI$ CANCEL OPERATION);
512      0648 2 $SCANEXH(DESBLK=GETQUIEXITBLOCK);
513      0649 2 IF .STATUS
514      0650 2      THEN STATUS = .IOSB[0];
515      0651 2
516      0652 2 IF NOT .STATUS
517      0653 2      THEN RETURN;
518      0654 2
519      0655 3 IF NOT (.QUEUE_FLAGS[QUI$V QUEUE FILE FLAG]
520      0656 3      OR .QUEUE_FLAGS[QUI$V_QUEUE FILE_FLAG_ONE])
521      0657 2      THEN .FLAG ITEM = SJCS_NO FILE FLAG;
522      0658 3 IF (.QUEUE_FLAGS[QUI$V QUEUE FILE TRAILER]
523      0659 3      OR .QUEUE_FLAGS[QUI$V_QUEUE FILE TRAILER_ONE])
524      0660 2      THEN .TRAILER_ITEM = SJCS_FILE_TRAILER_ONE;
525      0661 2 RETURN;
526      0662 1 END;
```

.EXTRN SYSSDCLEXH, SYSSCANEXH

001C 00000 GET_FLAG_TRAILER:

; Routine Size: 171 bytes, Routine Base: \$CODES + 045E

MAIL\$PRINT
V04-000

QUEUE MAIL MESSAGE FOR PRINTING

D 3
16-Sep-1984 01:14:12 VAX-11 Bliss-32 V4.0-742 Page 22
14-Sep-1984 12:42:29 DISK\$VMSMASTER:[MAIL.SRC]PRINT.B32;1 (10)

```

528 0663 1 GLOBAL ROUTINE MAIL$PRINT(QUEUE,COPIES,PRQ_FLAGS) : NOVALUE =
529 0664 1 ++
530 0665 1 FUNCTIONAL DESCRIPTION:
531 0666 1
532 0667 1 This routine is called to queue a copy of the current message
533 0668 1 for printing. The first time it is called, a job is
534 0669 1 created and the message is copied to a file that is added to the
535 0670 1 job. Each time it is called, and the queue name is the same as
536 0671 1 the last time, the current file will be added to the open job.
537 0672 1 If this routine is called with a different queue name, the previous
538 0673 1 job is closed, releasing it for printing, and a new job is opened.
539 0674 1
540 0675 1 If no queue name is specified, the queue specified in the previous
541 0676 1 call is assumed. If no queue name is specified the first call,
542 0677 1 SYSSPRINT is assumed.
543 0678 1
544 0679 1 An exit handler is declared on the first call to close the open
545 0680 1 job should this routine never be called with a different queue name.
546 0681 1
547 0682 1 INPUTS:
548 0683 1
549 0684 1 QUEUE = Address of specified queue name descriptor.
550 0685 1 COPIES = Number of copies, default is 1
551 0686 1
552 0687 1 OUTPUTS:
553 0688 1
554 0689 1 Temporary file created, message copied to it, print job created
555 0690 1 for specified queue, and temporary file added to job.
556 0691 1 --
557 0692 2 BEGIN
558 0693 2 MAP
559 0694 2 QUEUE: REF BLOCK[,BYTE], ! Queue name descriptor
560 0695 2 PRQ_FLAGS: BLOCK[,BYTE];
561 0696 2 LOCAL
562 0697 2 STATUS, ! Routine status
563 0698 2 PTR,
564 0699 2 FLAG_ITEM,
565 0700 2 TRAI[ER ITEM,
566 0701 2 CREJOBLIST: $ITMLST DECL(ITEMS=5),
567 0702 2 ADDFILELIST: $ITMLST DECL(ITEMS=5),
568 0703 2 ALTJOBLIST: $ITMLST DECL(ITEMS=3),
569 0704 2 JBCIOSB: VECTOR[2, LONG],
570 0705 2 RECDesc: $BBLOCK[DSC$C_S_BLN],
571 0706 2 MOREFLAG,
572 0707 2 TMP_FAB: $FAB DECL, ! Temp file FAB
573 0708 2 TMP_RAB: $RAB DECL, ! Temp file RAB
574 0709 2 TMP_NAM: $NAM DECL, ! Temp file NAM block
575 0710 2 TMP_ESNAM: BLOCK[NAM$C_MAXRSS, BYTE], ! Temp file name buffer
576 0711 2 BUFF: BLOCK[MAIL$R_INBUFSZ, BYTE]; ! Temp file I/O buffer
577 0712 2
578 0713 2 IF .FILES QUEUED NEQ 0
579 0714 2 AND .PRQ_FLAGS[PRQ_V_NOTIFY]
580 0715 3 THEN BEGIN
581 P 0716 3 $ITMLST INIT(ITMLST=ALTJOBLIST,
582 P 0717 3 (IT$COD=SJC$ QUEUE,
583 P 0718 3 BUFSTZ=.CURQUE_DESC[DSC$W_LENGTH],
584 P 0719 3 BUFADR=.CURQUE_DESC[DSC$A_POINTER]),

```

```
585 P 0720 3 (ITMCOD=SJCS_ENTRY_NUMBER,  
586 P 0721 3 BUFADR=CURJOBID),  
587 P 0722 3 (ITMCOD=SJCS_NOTIFY,  
588 P 0723 3 BUFSIZ=0  
589 P 0724 3 BUFADR=0$);  
590 P 0725 3 STATUS = $SNDJBCW(FUNC=SJCS.Alter_Job,  
591 P 0726 3 ITMLST=ALTJOBLIST,  
592 P 0727 3 IOSB=JBCIOSB);  
593 P 0728 3 IF NOT .STATUS  
594 P 0729 4 OR NOT (STATUS = .JBCIOSB[0])  
595 P 0730 3 THEN SIGNAL(MAIL$_ALTPRIJOB,2,.CURJOBID,CURQUE_DESC,.STATUS);  
596 P 0731 2 END;  
597 P 0732 2 If FORCE is specified, simply call close_job and return  
598 P 0733 2  
599 P 0734 2 IF .PRQ_FLAGS[PRQ_V_FORCE]  
600 P 0735 2 THEN IF .FILES queued EQL 0  
601 P 0736 2 THEN RETURN SIGNAL(MAIL$_NOMSGPRI)  
602 P 0737 2 ELSE RETURN CLOSE_JOB();  
603 P 0738 2  
604 P 0739 2 If there is no job currently open, or if a queue name is specified, and  
605 P 0740 2 it isn't the same as the current job's close the open job, and open  
606 P 0741 2 a new print job.  
607 P 0742 2  
608 P 0743 2  
609 P 0744 2 IF .CURQUE_DESC[DSC$W_LENGTH] EQL 0  
610 P 0745 3 OR (.QUEUE[DSC$W_LENGTH] NEQ 0  
611 P 0746 3 AND CH$NEQ(.QUEUE[DSC$W_LENGTH],.QUEUE[DSC$A_POINTER],  
612 P 0747 3 .CURQUE_DESC[DSC$W_LENGTH],.CURQUE_DESC[DSC$A_POINTER],0))  
613 P 0748 3 THEN BEGIN  
614 P 0749 3 CLOSE_JOB(); ! Close any open job  
615 P 0750 4 STR$COPY_DX(CURQUE_DESC,(IF .QUEUE[DSC$W_LENGTH] NEQ 0  
616 P 0751 4 THEN .QUEUE  
617 P 0752 3 ELSE $DESCRIPTOR('SYSSPRINT')));  
618 P 0753 3 GET FLAG TRAILER(FLAG_ITEM,TRAILER_ITEM);  
619 P 0754 3 SITMLST_INIT(ITMLST=CREJOBLIST,  
620 P 0755 3 (ITMCOD=SJCS_ENTRY_NUMBER_OUTPUT,BUFADR=CURJOBID),  
621 P 0756 3 (ITMCOD=SJCS_QUEUE,BUFSIZ=.CURQUE_DESC[DSC$W_LENGTH],  
622 P 0757 3 BUFADR=.CURQUE_DESC[DSC$A_POINTER]),  
623 P 0758 3 (ITMCOD=(IF .PRQ_FLAGS[PRQ_V_NOTIFY]  
624 P 0759 3 THEN SJCS_NOTIFY  
625 P 0760 3 ELSE SJCS_NO_NOTIFY),  
626 P 0761 3 BUFADR=0,  
627 P 0762 3 BUFSIZ=0$,  
628 P 0763 3 (ITMCOD=.FLAG_ITEM,BUFADR=0,BUFSIZ=0),  
629 P 0764 3 (ITMCOD=.TRAILER_ITEM,BUFADR=0,BUFSIZ=0));  
630 P 0765 3  
631 P 0766 3 STATUS = $SNDJBCW(FUNC=SJCS_CREATE_JOB,  
632 P 0767 3 ITMLST=CREJOBLIST,  
633 P 0768 3 IOSB=JBCIOSB);  
634 P 0769 3 IF NOT .STATUS  
635 P 0770 4 OR NOT (STATUS = .JBCIOSB[0])  
636 P 0771 4 THEN BEGIN  
637 P 0772 4 SIGNAL(MAIL$_CREPRIJOB,1,CURQUE_DESC,.STATUS);  
638 P 0773 4 RETURN STR$FREE1_DX(CURQUE_DESC); !No current queue  
639 P 0774 3  
640 P 0775 2 END;  
641 P 0776 2 !
```

```

642      0777 2 ! Create a temporary file
643      0778 2
644      P 0779 2 $FAB_INIT(      FAB = TMP_FAB,
645      P 0780 2             FNA = .LISTFILE_DESC[DSCSA_POINTER],
646      P 0781 2             FNS = .LISTFILE_DESC[DSCSW_LENGTH],
647      P 0782 2             CTX = MAIL$_OPENOUT,
648      P 0783 2             FOP = (TMP,500),
649      P 0784 2             NAM = TMP_NAM,
650      P 0785 2             RAT = (CR);
651      P 0786 2 $RAB_INIT(      RAB = TMP_RAB,
652      P 0787 2             FAB = TMP_FAB,
653      P 0788 2             CTX = MAIL$_OPENOUT,
654      P 0789 2             UBF = BUFF,
655      P 0790 2             USZ = MAIL$K_INBUFFSZ);
656      P 0791 2 $NAM_INIT(      NAM = TMP_NAM,
657      P 0792 2             RSA = TMP_ESNAM,
658      P 0793 2             RSS = NAM$C_MAXRSS,
659      P 0794 2             ESA = TMP_E5NAM,
660      P 0795 2             ESS = NAM$C_MAXRSS);
661      0796 2
662      0797 2 STATUS = $CREATE(FAB = TMP_FAB,ERR=UTIL$REPORT_IO_ERROR);! Create the file
663      0798 2 IF NOT .STATUS
664      0799 2     THEN RETURN .STATUS;
665      0800 2
666      0801 2 ! Copy the message to the file
667      0802 2
668      P 0803 2 IF _ERR($CONNECT(RAB = TMP_RAB,ERR=UTIL$REPORT_IO_ERROR);,
669      P 0804 2             TMP_FAB[FAB$V_DLT] = TRUE;
670      P 0805 2             $CLOSE(FAB=TMP_FAB);
671      P 0806 2             RETURN STATUS;
672      P 0807 2             TMP_RAB[RAB$L_CTX] = MAIL$_WRITEERR;
673      0808 2
674      0809 2 IF NOT .PRQ_FLAGS[PRQ_V_ALL]
675      0810 2     THEN STATUS = MAIL$EXTRACT_MESSAGE(1,TMP_RAB)
676      0811 2     ELSE IF .MAIL$GL_FLAGS[MAIF_V_ISAM]
677      0812 2         THEN STATUS = EXTRACT_A[L_ISAM(1,TMP_RAB)
678      0813 2         ELSE STATUS = EXTRACT_ALL_SEQ(1,TMP_RAB);
679      0814 2
680      0815 2 IF NOT .STATUS
681      0816 3     THEN BEGIN
682      0817 3             TMP_FAB[FAB$V_DLT] = 1; $CLOSE(FAB = TMP_FAB); ! Delete file
683      0818 3             RETURN;
684      0819 2             END;
685      0820 2             TMP_FAB[FAB$L_CTX] = MAIL$ CLOSEOUT;
686      P 0821 2 IF _ERR($CLOSE(FAB = TMP_FAB,ERR=UTIL$REPORT_IO_ERROR);, ! Close the file
687      0822 2             RETURN);
688      0823 2
689      0824 2 ! Add the file to the print job
690      0825 2
691      0826 2 PTR = CHSMOVE(NAMSS_DVI,TMP_NAM[NAM$T_DVI],BUFF);      ! Set device name
692      0827 2 PTR = CHSMOVE(NAMSS_FID,TMP_NAM[NAM$W_FID],.PTR);      ! Set file ID
693      0828 2 CHSMOVE(NAMSS_DID,TMP_NAM[NAM$W_DID],.PTR);          ! directory ID
694      P 0829 2 $ITMLST_INIT(ITMLST=ADDFILELIST
695      P 0830 2             (ITMCOD=SJCS_FILE IDENTIFICATION,
696      P 0831 2                 BUFSIZ=NAMSS_DVI+NAMSS_FID+NAMSS_DID,
697      P 0832 2                 BUFADR=BUFF),
698      P 0833 2             (ITMCOD=SJCS_DELETE_FILE,

```

```

699 P 0834 2           BUFADR=0,
700 P 0835 2           BUFSIZ=0},
701 P 0836 2           (ITMCOD=SJC$ FILE COPIES,
702 0837 2           BUFADR=COPIES));
703 P 0838 2           STATUS = $SNDJBCW(FUNC=SJC$ ADD FILE,
704 0839 2           ITM[ST=ADDFILELIST,
705 0840 2           IOSB=JBCIOSB);

706 0841 2           IF NOT .STATUS
707 0842 3           OR NOT (STATUS = .JBCIOSB[0])
708 0843 2           THEN SIGNAL(MAILS_CREPRIJOB,1,CURQUE_DESC,.STATUS)
709 0844 ELSE BEGIN
710 0845 3           FILES_QUEUED = .FILES_QUEUED + 1;
711 0846 3           IF NOT .EXITDECLARED           ! If no exit handler
712 0847 4           THEN BEGIN
713 0848 4           EXITDECLARED = $DCLEXH(DESBLK=EXITBLOCK);! Declare exit handler
714 0849 4           IF NOT .EXITDECLARED
715 0850 4           THEN
716 0851 4           SIGNAL(.EXITDECLARED);
717 0852 3           END:
718 0853 2           END:
719 0854 1 END;

```

54 4E 49 52 50 24 53 59 53 00509 P.AAR:	.ASCII \SYSSPRINT\	;
00000009 00512	.BLKB 2	;
00000000 00514 P.AAQ:	.LONG 9	;
00000000 00518	.ADDRESS P.AAR	;
	.EXTRN SYSSNDJBCW	
	OFFC 00000	
5B 0000000G 00 9E 00002	.ENTRY MAIL\$PRINT, Save R2,R3,R4,R5,R6,R7,R8,R9,-	0663
5A 0000000G 00 9E 00009	R10,R11	
59 0000000G 00 9E 00010	UTIL\$REPORT IO ERROR, R11	
58 DE AF 9E 00017	SYSSNDJBCW, RT0	
57 00000000 00 9E 0001B	LIB\$SIGNAL, R9	
5E FB4C CE 9E 00022	P.AAQ, R8	
14 A7 D5 00027	CURQUE DESC, R7	
60 13 0002A	-1204(SP), SP	
5C 0C AC E9 0002C	TSTL FILES_QUEUED	
50 FF58 CD 9E 00030	BEQL 2\$	
80 67 B0 00035	BLBC PRO FLAGS, 2\$	
80 86 8F 9B 00038	MOVAB ALTJOBLIST, \$\$ITMBLKPTR	0714
80 04 A7 D0 0003C	MOVAB SYSSNDJBCW, RT0	0724
80 80 D4 00040	MOVW CURQUE DESC, (\$\$ITMBLKPTR)+	
80 001E0004 8F D0 00042	MOVZBW #134, TSSITMBLKPTR)+	
80 10 A7 9E 00049	MOVL CURQUE DESC+4, (\$\$ITMBLKPTR)+	
80 80 D4 0004D	CLRL (\$\$ITMBLKPTR)+	
80 006C0000 8F D0 0004F	MOVL #1966084, (\$\$ITMBLKPTR)+	
80 80 7C 00056	MOVAB CURJOBID, (\$\$ITMBLKPTR)+	
80 80 D4 00058	CLRL (\$\$ITMBLKPTR)+	
7E 7E 7C 0005A	MOVL #7077888, (\$\$ITMBLKPTR)+	
FF50 CD 9F 0005C	CLRQ (\$\$ITMBLKPTR)+	
FF58 CD 9F 00060	CLRL (\$\$ITMBLKPTR)+	
7E 03 7D 00064	CLRQ -(SP)	0727
	PUSHAB JBCIOSB	
	PUSHAB ALTJOBLIST	
	MOVQ #3, -(SP)	

80	04	80	B4 00125	CLRW	(\$\$ITMBLKPTR)+			
		AE	B0 00127	MOVW	FLAG ITEM, (\$\$ITMBLKPTR)+			
		80	7C 0012B	CLRQ	(\$\$ITMBLKPTR)+			
		80	B4 0012D	CLRW	(\$\$ITMBLKPTR)+			
80		6E	B0 0012F	MOVW	TRAILER ITEM, (\$\$ITMBLKPTR)+			
		80	7C 00132	CLRQ	(\$\$ITMBLKPTR)+			
		80	D4 00134	CLRL	(\$\$ITMBLKPTR)+			
		7E	7C 00136	CLRQ	-(SP)	0768		
		FF50	CD 9F 00138	PUSHAB	JBCIOSB			
		CO	AD 9F 0013C	PUSHAB	CREJOBLIST			
		7E	0A 7D 0013F	MOVQ	#10, -(SP)			
			7E D4 00142	CLRL	-(SP)			
6A			07 FB 00144	CALLS	#7, SYSSNDJBCW			
56			50 D0 00147	MOVL	RO, STATUS	0769		
08			56 E9 0014A	BLBC	STATUS, 11\$	0770		
56		FF50	CD D0 0014D	MOVL	JBCIOSB, STATUS			
19			56 E8 00152	BLBS	STATUS, 12\$			
			56 DD 00155	PUSHL	STATUS	0772		
			57 DD 00157	PUSHL	R7			
			01 DD 00159	PUSHL	#1			
			8F DD 0015B	PUSHL	#8290354			
69			04 FB 00161	CALLS	#4, LIB\$SIGNAL			
			57 DD 00164	PUSHL	R7	0773		
00000000G	00		01 FB 00166	CALLS	#1, STR\$FREE1_DX			
			04 0016D	RET				
0050	8F	00	6E	00 2C 0016E	12\$:	MOVCS	#0, (SP), #0, #80, \$RMS_PTR	0785
			CD 00175					
		FEF8	CD 5003	8F B0 00178	MOVW	#20483, \$RMS_PTR		
		FEFC	CD 48	8F 9A 0017F	MOVZBL	#72, \$RMS_PTR+4		
		FF0E	CD	02 90 00185	MOVB	#2 \$RMS_PTR+22		
		FF10	CD 007E10A2	8F D0 0018A	MOVL	#8261794, \$RMS_PTR+24		
		FF16	CD 0202	8F B0 00193	MOVW	#514, \$RMS_PTR+30		
		FF20	CD FE54	CD 9E 0019A	MOVAB	TMP_NAM, \$RMS_PTR+40		
		FF24	CD FB08	C8 D0 001A1	MOVL	LISTFILE_DESC+4, \$RMS_PTR+44		
		FF2C	CD FB04	C8 90 001A8	MOVB	LISTFILE_DESC, \$RMS_PTR+52		
0044	8F	00	6E	00 2C 001AF	MOVCS	#0, (SP), #0, #68, \$RMS_PTR		0790
			CD 001B6					
		FEB4	CD 4401	8F B0 001B9	MOVW	#17409, \$RMS_PTR		
		FECC	CD 007E10A2	8F D0 001C0	MOVL	#8261794, \$RMS_PTR+24		
		FED4	CD 0200	8F B0 001C9	MOVW	#512, \$RMS_PTR+32		
		FED8	CD 08	AE 9E 001D0	MOVAB	BUFF, \$RMS_PTR+36		
0060	8F	00	FEF0	CD FEF8	CD 9E 001D6	MOVAB	TMP_FAB, \$RMS_PTR+60	
			00 2C 001DD	MOVCS	#0, -(SP), #0, #96, \$RMS_PTR		0795	
			CD 001E4					
		FE54	CD 6002	8F B0 001E7	MOVW	#24578, \$RMS_PTR		
		FE56	CD	01 8E 001EE	MNEG_B	#1, \$RMS_PTR+2		
		FE58	CD 0208	CE 9E 001F3	MOVAB	TMP_ESNAM, \$RMS_PTR+4		
		FE5E	CD	01 8E 001FA	MNEG_B	#1, \$RMS_PTR+10		
		FE60	CD 0208	CE 9E 001FF	MOVAB	TMP_ESNAM, \$RMS_PTR+12		
			5B DD 00206	PUSHL	R11	0797		
		FEF8	CD 9F 00208	PUSHAB	TMP_FAB			
00000000G	00		02 FB 0020C	CALLS	#2, SYSSCREATE			
		56	50 D0 00213	MOVL	RO, STATUS	0798		
		01	56 E8 00216	BLBS	STATUS, 13\$			
			04 00219	RET				
		FEB4	5B DD 0021A	13\$:	PUSHL	R11	0806	
			CD 9F 0021C	PUSHAB	TMP_RAB			

K 3
16-Sep-1984 01:14:12
14-Sep-1984 12:42:29VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]PRINT.B32;1Page 29
(11)

00000000G	00	02	FB 00220	CALLS	#2, SYSSCONNECT		
52	50	DO 00227	MOVL	R0, STATUS			
40	52	E9 0022A	BLBC	STATUS, 17\$			
OD	FECC	CD 007E10D2	8F	DO 0022D	MOVL	#8261842, TMP_RAB+24	0807
OC	AC	FEB4	02	E0 00236	BBS	#2, PRQ_FLAGS, 14\$	0809
			01	DD 0023F	PUSHAB	TMP_RAB	0810
F92E	CF		02	FB 00241	PUSHL	#1	
			1F	11 00246	CALLS	#2, MAIL\$EXTRACT_MESSAGE	
		OD 00000000G	00	E9 00248	BRB	16\$	
		FEB4	CD	9F 0024F	BLBC	MAIL\$GL_FLAGS, 15\$	0811
			01	DD 00253	PUSHAB	TMP_RAB	0812
F9C1	CF		02	FB 00255	PUSHL	#1	
			0B	11 0025A	CALLS	#2, EXTRACT_ALL_ISAM	
			CD	9F 0025C	BRB	16\$	
		FEB4	01	DD 00260	PUSHAB	TMP_RAB	0813
F9EB	CF		02	FB 00262	PUSHL	#1	
56	50		50	DO 00267	CALLS	#2, EXTRACT_ALL_SEQ	
12	56		56	E8 0026A	MOVL	R0, STATUS	
FEFD	CD	80	8F	88 0026D	BLBS	STATUS, 18\$	0815
		FEF8	CD	9F 00273	BISB2	#128, TMP_FAB+5	0817
00000000G	00		01	FB 00277	PUSHAB	TMP_FAB	
			04	0027E	CALLS	#1, -SYSSCLOSE	
			FF10	CD 007E105A	RET		
			5B	DO 0027F	MOVL	#8261722, TMP_FAB+24	0816
		FEF8	CD	9F 0028A	PUSHL	R11	0820
00000000G	00		02	FB 0028E	PUSHAB	TMP_FAB	0822
			50	E8 00295	CALLS	#2, -SYSSCLOSE	
			04	00298	BLBS	STATUS, 19\$	
					RET		
08	AE	FE68	CD	10 28 00299	MOV3	#16, TMP_NAM+20, BUFF	0826
63	FE78	CD	06	28 002A0	MOV3	#6, TMP_NAM+36, (PTR)	0827
63	FE7E	CD	06	28 002A6	MOV3	#6, TMP_NAM+42, (PTR)	0828
		50	AD 9E 002AC	MOVAB	ADDFILE[IST, \$\$ITMBLKPTR		
		80 0027001C	8F DO 002B0	MOVL	#\$2555932, (\$\$ITMBLKPTR)+		
		80 08	AE 9E 002B7	MOVAB	BUFF, (\$\$ITMBLKPTR)+		
		80 00180000	80 D4 002BB	CLRL	(\$\$ITMBLKPTR)+		
		80 00230004	80 D0 002BD	MOVL	#1572864, (\$\$ITMBLKPTR)+		
		80 08	AC 9E 002C4	CLRQ	(\$\$ITMBLKPTR)+		
			80 D0 002C6	MOVL	#2293764, (\$\$ITMBLKPTR)+		
			AC 9E 002CD	MOVAB	COPIES, (\$\$ITMBLKPTR)+		
			80 7C 002D1	CLRQ	(\$\$ITMBLKPTR)+		
			7E 7C 002D3	CLRQ	-(SP)		
		FF50	CD 9F 002D5	PUSHAB	JBCIOSB	0840	
		80	AD 9F 002D9	PUSHAB	ADDFILELIST		
		7E	02 7D 002DC	MOVQ	#2, -(SP)		
		7E	D4 002DF	CLRL	-(SP)		
		6A	07 FB 002E1	CALLS	#7, SYSSNDJBCW		
		56	50 DO 002E4	MOVL	R0, STATUS		
		08	56 E9 002E7	BLBC	STATUS, 20\$	0841	
		56	CD 002EA	MOVL	JBCIOSB, STATUS	0842	
		10	56 E8 002EF	BLBS	STATUS, 21\$	0843	
			56 DD 002F2	PUSHL	STATUS		
			57 DD 002F4	PUSHL	R7		
			01 DD 002F6	PUSHL	#1		
		007E8032	8F DD 002F8	PUSHL	#8290354		
69			04 FB 002FE	CALLS	#4, LIB\$SIGNAL		
			04 00301	RET			

MAIL\$PRINT
V04-000

QUEUE MAIL MESSAGE FOR PRINTING

L 3

16-Sep-1984 01:14:12
14-Sep-1984 12:42:29

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]PRINT.B32;1

Page 30
(11)

		14	A7	D6	00302	21\$:	INCL	FILES QUEUED	0845
	16	3C	A7	E8	00305		BLBS	EXITDECLARED, 23\$	0846
00000000G	00	2C	A7	9F	00309		PUSHAB	EXITBLOCK	0848
3C	A7	01	FB	0030C			CALLS	#1, SYSSDCLEXH	
05	50	50	DD	00313			MOVL	R0, EXITDECLARED	0849
	69	50	E8	00317			BLBS	R0, 23\$	0851
		01	DD	0031A			PUSHL	R0	
		04	FB	0031C	22\$:		CALLS	#1, LIB\$SIGNAL	
				0031F	23\$:		RET		0854

; Routine Size: 800 bytes, Routine Base: \$CODE\$ + 051C

```
721 0855 1 ROUTINE CLOSE_JOB : NOVALUE =
722 0856 1 ++
723 0857 1 FUNCTIONAL DESCRIPTION:
724 0858 1
725 0859 1 This routine is called to close any currently open print job,
726 0860 1 releasing it for printing.
727 0861 1
728 0862 1 It is also called as an exit handler at image run down.
729 0863 1
730 0864 1 INPUTS:
731 0865 1
732 0866 1 CURQUEUE = Current queue name.
733 0867 1 = zero if none.
734 0868 1
735 0869 1 OUTPUTS:
736 0870 1
737 0871 1 CURQUEUE = zero.
738 0872 1
739 0873 1 If a print job is open, it is closed.
740 0874 1 ---
741 0875 2 BEGIN
742 0876 2
743 0877 2 LOCAL
744 0878 2 STATUS,
745 0879 2 CLOSEJOBLIST : $ITMLST DECL(ITEMS=1),
746 0880 2 JBCSTSBUF : $BBLOCK[NAM$C_MAXRSS],
747 0881 2 DESC : VECTOR[2, LONG],
748 0882 2 JBCIOSB : VECTOR[2, LONG];
749 0883 2
750 0884 2 IF .CURQUEUE DESC[DSC$W_LENGTH] NEQ 0
751 0885 3 THEN BEGIN
752 P 0886 3 $ITMLST_INIT(ITMLST=CLOSEJOBLIST,
753 P 0887 3 (ITMCOD=SJC$ JOB STATUS OUTPUT,
754 P 0888 3 BUFSTZ=NAM$C MAXRSS,
755 P 0889 3 BUFADR=JBCSTSBUF,
756 P 0890 3 RETLEN=DESC[0]));
757 0891 3 DESC[0] = 0;
758 0892 3 DESC[1] = JBCSTSBUF;
759 P 0893 4 IF (STATUS = $SNDJBCW(FUNC=SJC$ CLOSE_JOB,
760 P 0894 4 ITMLST=CLOSEJOBLIST,
761 P 0895 4 IOSB=JBCIOSB))
762 0896 4 AND (STATUS = .JBCIOSB[0])
763 0897 4 THEN BEGIN
764 0898 4 IF .FILES queued NEQ 0
765 0899 4 THEN MAIL$PUT_OUTPUT(DESC);
766 0900 4 END
767 0901 3 ELSE IF .FILES queued NEQ 0
768 0902 3 THEN SIGNAL(MAIL$ CREPRIJOB,1,CURQUEUE_DESC,,STATUS);
769 0903 3 STR$FREE1 DX(CURQUEUE_DESC); !No current queue
770 0904 3 FILES queued = 0;
771 0905 2 END;
772 0906 1 END;
```

0004 00000 CLOSE_JOB:

52 00000000'	00 9E 00002	.WORD	Save R2	0855
5E FEE0	CE 9E 00009	MOVAB	CURQUE DESC, R2	
	62 B5 0000E	MOVAB	-288(SP), SP	
	6F 13 00010	TSTW	CURQUE_DESC	0884
50 F0	AD 9E 00012	MOVAB	CLOSEJOBLIST, \$\$ITMBLKPTR	0890
80 005800FF	8F D0 00016	MOVL	#5767423, (\$\$ITMBLKPTR)+	
80 10	AE 9E 0001D	MOVAB	JBCSTSBUF, (\$\$ITMBLKPTR)+	
80 08	AE 9E 00021	MOVAB	DESC, (\$\$ITMBLKPTR)+	
	80 D4 00025	CLRL	(\$\$ITMBLKPTR)+	
OC AE	08 AE D4 00027	CLRL	DESC	0891
	10 AE 9E 0002A	MOVAB	JBCSTSBUF, DESC+4	0892
	08 AE 9F 00031	CLRQ	-(SP)	0895
	F0 AD 9F 00034	PUSHAB	JBCIOSB	
7E	09 7D 00037	PUSHAB	CLOSEJOBLIST	
	7E D4 0003A	MOVQ	#9, -(SP)	
00000000G 00	07 FB 0003C	CLRL	-(SP)	
17	50 E9 00043	CALLS	#7, SYSSNDJBCW	
50	6E D0 00046	BLBC	STATUS, 1\$	
11	50 E9 00049	MOVL	JBCIOSB, STATUS	0896
	14 A2 D5 0004C	BLBC	STATUS, 1\$	
	24 13 0004F	TSTL	FILES_QUEUED	0898
00000000G 00	08 AE 9F 00051	BEQL	2\$	
	01 FB 00054	PUSHAB	DESC	0899
	18 11 0005B	CALLS	#1, MAIL\$PUT_OUTPUT	
	14 A2 D5 0005D	1\$: TSTL	2\$	0893
	13 13 00060	BEQL	FILES_QUEUEL	0901
	50 DD 00062	PUSHL	STATUS	
	52 DD 00064	PUSHL	R2	0902
00000000G 00	01 DD 00066	PUSHL	#1	
007E8032	8F DD 00068	PUSHL	#8290354	
00000000G 00	04 FB 0006E	CALLS	#4, LIB\$SIGNAL	
	52 DD 00075	2\$: PUSHL	R2	0903
	01 FB 00077	CALLS	#1, STR\$FREE1_DX	
	14 A2 D4 0007E	CLRL	FILES_QUEUED	0904
	04 00081	3\$: RET		0906

; Routine Size: 130 bytes, Routine Base: \$CODE\$ + 083C

MAIL\$PRINT
VO4-000

QUEUE MAIL MESSAGE FOR PRINTING

8 4
16-Sep-1984 01:14:12
14-Sep-1984 12:42:29
VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]PRINT.B32;1 Page 33
(13)

: 774 0907 1 END
: 775 0908 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	72	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODES	2238	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Symbols -----	Pages	Processing
	Total Loaded Percent	Mapped	Time
\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776 193 1	581	00:00.7
\$255\$DUA28:[MAIL.OBJ]MAILDEF.L32;1	457 11 2	26	00:00.2

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:PRINT/OBJ=OBJ\$:PRINT MSRC\$:PRINT/UPDATE=(ENH\$:PRINT)

Size: 2100 code + 210 data bytes
Run Time: 00:30.5
Elapsed Time: 01:50.5
Lines/CPU Min: 1789
Lexemes/CPU-Min: 48975
Memory Used: 332 pages
Compilation Complete

0231 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

READ
LIS

SENDMSG
LIS

PRINT
LIS

SPAWNCMD
LIS

MANAGE

ALFMaint
LIS

SET
LIS

SYSMAINT
LIS

SHOW
LIS